(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 084 270
A3

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 82307026.3

(51) Int. Cl.³: H 04 N 7/12

(22) Date of filing: 31.12.82

(30) Priority: 04.01.82 US 336984

(43) Date of publication of application: 27.07.83 Bulletin 83/30

(88) Date of deferred publication of search report: 31.08.83

(84) Designated Contracting States: DE FR GB

(71) Applicant: COMPRESSION LABS, INC.
2305 Bering Drive
San Jose California 95131(US)

(72) Inventor: Hamilton, Eric R.
137 El Altillo
Los Gatos California 95030(US)

(72) Inventor: Fralick, Stanley C.
21476 Sarahills Ct.
Saratoga California 95070(US)

(72) Inventor: Widergren, Robert D.
19397 Zinfandel Ct.
Saratoga California 95070(US)

(72) Inventor: Tescher, Andrew G.
925 Maryhurst Dr.
Claremont California 91711(US)

(74) Representative: Crawford, Andrew Birkby et al.
A.A. THORNTON & CO. Northumberland House 303-306
High Holborn
London WC2A 1AY(GB)

(54) Video bandwidth reduction system employing interframe block differencing and transform domain coding.

(57) Video type information signals are compressed by comparing corresponding blocks of time domain information signals from successive fields, converting a block of these signals to a transform domain signal represented by descrete cosine transform coefficients when the difference between the corresponding blocks exceeds a block difference threshold, and encoding the transform domain coefficients for transmission. Corresponding blocks are compared by storing the successive fields in memory on a pixel by pixel basis and forming the difference between corresponding pixels from the successive blocks. Blocks are converted by first transforming individual samples along the horizontal direction, and then transforming the same block samples along the vertical direction, the transformed coefficients being stored in a diagonal memory unit. Each converted block stored in the diagonal memory is encoded using a plurality of unique code tables. The block codes are assembled in a transmission buffer in the order of generation and are transmitted to a decoding site.

./...

EP 0 084 270 A3

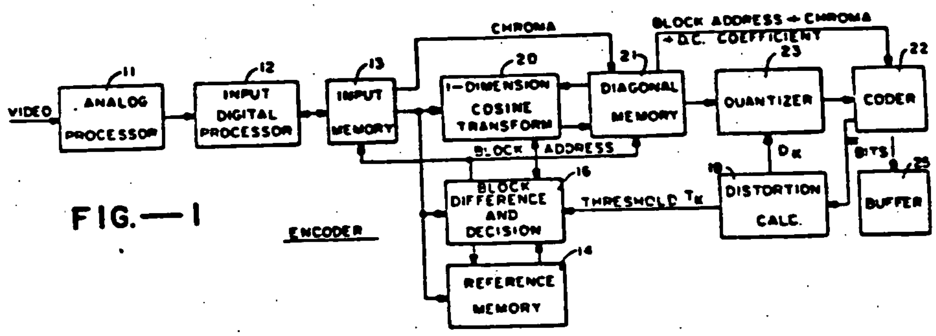

FIG.—1

ENCODER

(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: 0 084 270
A2

EUROPEAN PATENT APPLICATION

(21) Application number: 82307026.3

(22) Date of filing: 31.12.82

(51) Int. Cl.[3]: H 04 N 7/12

(30) Priority: 04.01.82 US 336984

(43) Date of publication of application: 27.07.83 Bulletin 83/30

(84) Designated Contracting States: DE FR GB

(71) Applicant: COMPRESSION LABS, INC.
2305 Bering Drive
San Jose California 95131(US)

(72) Inventor: Hamilton, Eric R.
137 El Altillo
Los Gatos California 95030(US)

(72) Inventor: Fralick, Stanley C.
21476 Sarahills Ct.
Saratoga California 95070(US)

(72) Inventor: Widergren, Robert D.
19397 Zinfandel Ct.
Saratoga California 95070(US)

(72) Inventor: Tescher, Andrew G.
925 Maryhurst Dr.
Claremont California 91711(US)

(74) Representative: Crawford, Andrew Birkby et al,
A.A. THORNTON & CO. Northumberland House 303-306 High Holborn
London WC2A 1AY(GB)

(54) **Video bandwidth reduction system employing interframe block differencing and transform domain coding.**

(57) Video type information signals are compressed for transmission and reproduction by comparing corresponding blocks of time domain information signals from successive fields, converting a block of the time domain information signals to a transform domain signal represented by discrete cosine transform coefficients when the difference between the corresponding blocks exceeds a block difference threshold, and encoding the transform domain coefficients for transmission to a decoding site. Corresponding blocks of time domain information signals from successive fields are compared by storing the successive fields in memory on a pixel by pixel basis, retrieving each block on a pixel by pixel basis, forming the difference between corresponding pixels from the successive blocks, squaring the resulting difference signal, summing the squares and dividing by the number of pixels per block. Successive fields are merged by weighted summing of corresponding pixels.

Blocks are converted by first transforming individual samples along the horizontal direction, and then transforming the same block samples along the vertical direction, the transformed coefficients being stored in a diagonal memory unit along with a block address code and block averaged quadrature chrominance characters. Each converted block stored in the diagonal memory is encoded using a plurality of unique code tables. A dedicated block address code table is constructed using a unique algorithm; the remaining code tables are Huffman coded tables, one dedicated to the D.C. transform coefficient term, two other tables dedicated to the quadrature component characters, and the remaining tables (including the two chroma tables) being selected on a coefficient by coefficient basis using an algorithm based on the predictive mean value of the cosine coefficient terms. The cosine coefficient terms are also quantized prior to encoding by dividing the coefficient by a variable parametric value D_k, which is a measure of the fullness of the transmission buffer, and which is changed in value after each

./...

EP 0 084 270 A2

block of codes is loaded into the buffer. The D_k value is also used to adjust the value of the block difference threshold.

The block codes are assembled in a transmission buffer in the order of generation and are transmitted to a decoding site along with initialization values for D_k and a buffer fullness parameter a_k.

At the decoding site, the received code characters are decoded using an inverse processing resulting in recapture of the initial video signals.

The luminance and chrominance components are initially processed by subsampling and averaging to produce further compression.

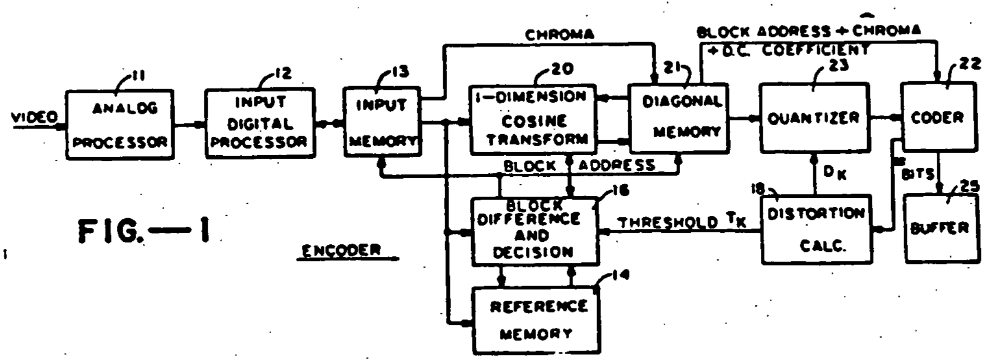

FIG.—1

VIDEO BANDWITH REDUCTION SYSTEM EMPLOYING INTERFRAME BLOCK DIFFERENCING AND TRANSFORM DOMAIN CODING

BACKGROUND OF THE INVENTION

This invention relates to information signal processing in general, and in particular to the field of processing time sequential information signals (such as video signals) for the purpose of compressing the amount of information to be transferred from an encoding site to a decoding site.

In recent years, increasing efforts have been directed toward providing more efficient information signal encoding techniques used to process time sequential information signals prior to their transmission from a transmitting station to a receiving station. The requirement for more efficient encoding techniques has been prompted by two major factors: firstly, a substantial increase in the quantity of information required to be transferred via communication links and, secondly, maximum occupancy of the communication frequency bands available for voice and data transmission. An early technique employed to reduce the amount of information required to be transferred without substantial degradation is the signal processing technique known as conditional replenishment, described in U.S. Patent No. 3,984,626 to Mounts et al., the disclosure of which is hereby incorporated by reference. Briefly, in the conditional replenishment signal processing technique, individual line element sample signals from a successive field of information are compared with the corresponding line elements in the previous field, and the difference therebetween is tested against a fixed threshold. If the difference exceeds the threshold value, the new value is encoded and transmitted to a receiving station, along with an appropriate address code specifying the line location of the sample to be updated in the field memory of the receiving station. Thus, rather than transmitting each and every line

sample for every field, only those samples which differ by a significant threshold amount are transmitted, which substantially reduces the number of samples in the communication channel pipeline. Although this saving in the amount of actual data flowing through the communication pipeline is somewhat offset by the necessity of simultaneously transmitting the address information, this disadvantage is more than overcome by the substantial reduction in the total number of samples which must be transmitted in order to maintain the information current at the decoding site. When used to process video type information signals, an even greater reduction in the required number of transmitted samples is achieved due to the inherent nature of video signals, which possess intrinsic interfield correlation (e.g. abrupt interfield changes for background portions of video images occur relatively infrequently).

Another compression technique known in the art is the use of transform domain encoding, in which each field of information signals is divided into a number of rectangular or square arrays of individual picture elements (for example a 16 pixel by 16 pixel array) termed blocks, and each block is converted to the transform domain. For each converted block, the individual transform coefficients are then encoded and transmitted along with appropriate address codes, as well as additional overhead information (e.g. field start signals, frame start signals and the like). One such transform domain processing system is disclosed in U.S. Patent No. 4,189,748 to Reis, the disclosure of which is hereby incorporated by reference.

Although many types of mathematical transform functions have been proposed for implementation in a transform domain signal processing system, in reality most transform functions are inappropriate for implementation due to the complexity of the required logic circuitry. This disadvantage is exacerbated in applications requiring real time signal processing by virtue of the minimum time period required to perform the signal processing necessary to generate the values of the transform coefficients. For a

general discussion of the advantages and disadvantages of the different types of transform functions, reference should be had to the collection of technical publications entitled "Image Transmission Techniques, Advances in Electronics and Electron Physics, Supplement 12", Pratt, Academic Press, 1979, particularly the section entitled "Transform Image Coding".

SUMMARY OF THE INVENTION

The invention comprises a method and system for processing time domain information signals which combines the advantages of conditional replenishment and transform domain coding in such a manner that information signal compression of a magnitude substantially greater than that available in known systems is achieved while affording real time information signal processing.

In its broadest aspect, the invention provides a method of processing time domain information signals having a successive field format to effect substantial compression of the signals, the method including the steps of comparing corresponding blocks of time domain information signals from successive fields, converting a block of the time domain information signals to a transform domain signal represented by discrete cosine transform coefficients when the difference between the corresponding blocks exceeds a first variable parametric value, and encoding the transform domain coefficients for subsequent utilization, e.g. transmission from a transmitting station to a receiving station, recording on video tape or other magnetic media, etc. The corresponding blocks of time domain information signals from successive fields are compared by storing the successive fields in memory on a pixel by pixel basis, retrieving the corresponding blocks from memory also on a pixel by pixel basis, forming the difference between corresponding pixels from the successive blocks, squaring the resulting difference signal, summing the squares of the resulting difference signals, and dividing the resulting sum by the number of pixels per block. In the preferred embodiment, the method is optimized by employing a total of 64 pixels per block

arranged in an 8 by 8 array and by merging successive fields on a pixel by pixel basis, the merging being performed by summing corresponding pixels from successive fields in accordance with a predetermined weighting factor of 3/4 for the earlier appearing (previously merged) field and 1/4 for the later appearing field.

The conversion of a block of the time domain information signals to the transform domain is accomplished by first transforming the individual block samples along a first direction, which is the horizontal line direction in the preferred embodiment, and subsequently transforming the same block samples along the orthogonal direction, which is the vertical direction in the preferred embodiment. For each transformed block, the individual block samples corresponding to the previous field are replaced with the updated block information, and the transformed coefficients for the converted block are stored in diagonal format in a diagonal memory unit. In addition, an address code indicating the field address of a transformed block is also stored in the diagonal memory for subsequent encoding.

The transform coefficients for each converted block stored in the diagonal memory are encoded using a plurality of different code tables, one of the tables being dedicated to the first coefficient in each diagonal group, corresponding to the DC term and representing the average signal intensity of the converted block, and the remaining tables being selected on a coefficient by coefficient basis. Specifically, each transform coefficient (other than the first or DC coefficient) is first quantized by digitally dividing the coefficient by a variable parametric value D_K, after which the predictive mean value of each quantized coefficient is calculated by summing a weighted portion of the actual value of that quantized coefficient with the predictive mean value of the previous quantized coefficient weighted by a different factor, and the newly calculated predictive mean value is used to select that one of the several available individual code tables capable of encoding the quantized coefficient value with a minimum number of

binary bits. In addition to the tables noted above, separate tables are provided for encoding the block address of the encoded transform coefficients, for directly encoding the D.C. coefficient, and for run length coding certain coefficient values. In the case of time domain information signals comprising color video signals with quadrature components two preselected quantized coefficient code tables are used to represent the average value of each color quadrature component of the corresponding converted block.

To further compress the amount of information encoded prior to utilization, those successive transform coefficients with zero value whose predictive mean lies below the value of a preselected fixed threshold are transmitted as a run length code. In addition, when the predictive values for successive remaining cosine coefficients in the converted block lie below the preselected fixed threshold, a single end-of-block code is generated.

The codes corresponding to a given converted block are transferred at a variable rate to a rate buffer in the order of generation prior to utilization, and the number of binary bits transferred to the buffer is monitored in order to gauge the buffer fullness. The dynamic occupancy of the buffer is used to control the value of the variable parametric value D_K in order to minimize the possibility of buffer overflow, utilizing a special algorithm. The buffer fullness state is also used to control the first variable parametric value -- termed the block difference threshold T_K -- also by employing a special algorithm. Thus, as the rate buffer approaches the completely filled state, the magnitude of D_K is increased, which increases the minimum quantizing interval employed in sampling the transform coefficients during the encoding process. In addition, the block difference threshold T_K is similarly increased to reduce the number of blocks selected for conversion to the transform domain and subsequent encoding. Similarly, as the state of the buffer fullness decreases, both D_K and T_K are lowered in value in accordance with the special algorithms

employed in order to increase the number of blocks selected for conversion to the transform domain and to decrease the minimum quantization interval used in the encoding process.

The codes representing the converted blocks are formatted in the rate buffer in the following fashion. The start of each frame is denoted by a frame sync code signal, which is followed by a first control code signal representative of the buffer fullness at the beginning of the frame and a second control code signal representative of the quantizing interval D_K value at the beginning of the same frame. The control code signals are followed by individual block replenishment code symbols which include a block address code specifying the field address of the corresponding block, the DC code term representative of the average intensity of the corresponding block, and the plurality of coefficient code terms representative of the predictive mean value of the transform coefficients for the corresponding block. For color video signal processing, the quadrature component code terms are included between the block address code and the DC code term. The termination of the last block is signified by the subsequent appearance of the frame sync code signal for the next succeeding frame.

The decoding process is essentially the inverse of the encoding process. For each frame of encoded information, the first and second control code signals are used to establish the initial minimum quantization interval to be employed for inverse quantizing the block replenishment code symbols. The received replenishment code symbols are decoded using a parallel set of inverse code tables, which are selected using the same predictive mean algorithm as that employed in the encoding process. The block address, quadrature chrominance and D.C. term codes are coupled directly to a diagonal memory unit, while the coefficient code terms are inverse quantized by multiplying each code term by D_K, using the transmitted initial value of D_K for the first block of data, and the resulting coefficients are stored in the diagonal memory unit. After the first block has been decoded, the distortion constant D_K is recalculated

and the newly calculated value of D_k is used to inverse quantize the next block of data.

The coefficients stored in the diagonal memory unit are then transformed to time domain digital samples using an inverse discrete cosine transform, and the resulting samples are stored in an output memory unit, replacing previous samples representing the same block. The merged field samples stored in the output memory unit, which replicate the merged field samples stored in a corresponding reference memory unit at the encoder site, are finally processed to provide video output signals.

Further compression is achieved according to the invention by special initial processing of the luminance and chrominance samples. The luminance signals are sub-sampled at less than the standard rate (which is 512 lines/frame and 512 samples/line for NTSC video), the preferred embodiment employing 256 lines/frame and 256 samples/line. Each quadrature chrominance component is sub-sampled at less than the standard rate and averaged over a given block. In the preferred embodiment, each quadrature component is sub-sampled at one-half the standard rate for each block line and the sub-samples for each block line are averaged, after which each block line average is combined to obtain a block average. Prior to averaging, each chrominance component sample is further modified by discarding the two least significant bits of the sample. After transmission from an encoding site to a decoding site, the full range of luminance and chrominance samples is recovered by individual interpolative processing of the received luminance and chrominance samples.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram illustrating an encoder incorporating the invention;

Fig. 2 is a block diagram illustrating a decoder incorporating the invention;

Fig. 3 is a schematic view of a portion of a display screen illustrating the replenishment block size;

Fig. 4 is a trellis diagram illustrating the cosine transform alogorithm employed in the preferred embodiment;

Fig. 5 is a schematic diagram illustrating the manner in which transform coefficients are stored in a diagonal memory unit;

Fig. 6 is set of probability distribution curves illustrating the manner in which the quantized coefficient encoding tables are constructed;

Fig. 7 is a schematic diagram illustrating typical predictive mean values for a single block; and

Fig. 8 is a schematic diagram illustrating the code formatting for one frame of replenishment information.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

Turning now to the drawings, Fig. 1 is a block diagram illustrating a preferred embodiment of the encoder portion of the invention. As seen in this Fig., analog video signals are coupled to the input of an analog processor unit 11 in which composite video input signals are separated into the standard luminance and quadrature chrominance components and converted to multi-bit digital samples at a predetermined sampling rate. In the preferred embodiment eight bit digital samples are taken at a 10.7 MHz sampling rate. The equivalent digital data samples produced in analog processor unit 11 are coupled to the input of an input digital processor 12 in which incoming field samples are merged with the corresponding samples from the previous field in the manner described below. The resulting individual merged field samples from input digital processor 12 are stored in an input memory unit 13 having a sufficient capacity to contain one field of digital information. An additional memory unit 14, termed a reference memory, is coupled to the data output of input memory unit 13. Reference memory unit 14 stores a reference field of information for comparison with a newly merged field stored in input memory unit 13, and has the same capacity as input memory unit 13.

In operation, the individual digital samples from an incoming field supplied to input digital processor 12 are

added to the corresponding digital samples of the previously merged field stored in input memory unit 13 on a weighted basis, and the resulting weighted sums are stored in input memory unit 13, replacing the previously stored samples on a pixel by pixel basis. In the preferred embodiment, the samples are weighted by a factor of 3 to 1 between the older samples stored in the input memory unit 13 and the incoming field samples, i.e. the earlier samples are multiplied by a factor of 3/4, the later samples are multiplied by the factor of 1/4 and the resulting weighted samples are added together. The weighting multiplication and the addition are accomplished with conventional digital multipliers and adders, in combination with appropriate conventional addressing logic.

Each newly merged field stored in input memory unit 13 is compared on a block by block basis with the reference field stored in reference memory unit 14 by means of a block difference and decision unit 16. As illustrated in Fig. 3, each block element consists of a rectangular array of 8 pixels by 8 pixels, and the difference between each block is obtained by digitally subtracting corresponding pixel samples read from input memory unit 13 and reference memory unit 14 in block difference and decision unit 16, squaring the resulting difference signals, summing the squares of the resulting difference signals, and dividing the resulting sum by the number of pixels per block (64). Each block difference value so obtained is tested against a threshold T_K supplied from a distortion calculation unit 18. If the block difference exceeds the threshold, the corresponding block in input memory unit 13 is converted to a set of transform coefficients by means of a one dimension cosine transform unit 20, and the transform coefficients are stored in a diagonal memory unit 21 along with a corresponding block address code specifying the field block to which the transform coefficients correspond. In addition, whenever a block is selected for conversion to the transform domain, the reference memory unit 14 is updated by replacing the

corresponding block in reference memory unit 14 with the newly selected block.

The conversion of each selected block to the transform domain is done by a one dimensional cosine transform unit 20 in two steps: a first transformation along the horizontal direction, followed by a second transformation along the vertical direction. The cosine transform unit 20 implements the well known discrete cosine transform function:

$$F(R) = \frac{2C(k)}{N} \sum_{j=0}^{N-1} f(j)\cos\left[\frac{(2j+1)k\pi}{2N}\right]$$

where $C(k) = \frac{1}{\sqrt{2}}$ at zero, 1 for k (1,N-1) and zero elsewhere

and comprises a plurality of conventional digital multiplying accumulators configured to implement the 8 point cosine transform algorithm shown in the trellis diagram of Fig. 4. During the transformation along the horizontal direction, the developing coefficients are stored in diagonal memory unit 21, and are subsequently recalled during the transformation in the vertical direction. After the selected block has been completely converted to the transform domain, the resulting series of coefficients is stored in diagonal memory unit 21 along with a multi-bit digital word specifying the block address of the block corresponding to the series of coefficients and two multi-bit digital words specifying the average value of the chrominance quadrature components for the corresponding block, the coefficients being arranged in the diagonal form illustrated schematically in Fig. 5.

The transform coefficients a[illegible] corresponding block address and chrominance quadrature dig[illegible]al characters are next encoded for subsequent transmission in the following manner. The block address digital character corresponding to a series of transform coefficients is coupled directly to a coder unit 22 which contains in the preferred embodiment nine separate code tables, eight tables containing a set of

code characters arranged according to the Huffman code technique, in which the number of bits per specific character depends upon the probability of occurrence of that character, and one table containing a set of code characters arranged according to a special variable length coding technique specified below. The special variable length code table is dedicated for use with the block address code, and the application of a new block address code to the dedicated code table results in the generation of a block address transmission code. The block address is actually encoded by forming the numerical difference between the current block address and the address of the most recent previously encoded block address, and generating a code in accordance with the following algorithm:

If $\Delta_k = 1$, code 1 bit (Δ_k)

If $\Delta_k < 32$, code 1 bit zero + 5 bits (Δ_k)

If $\Delta_k \geq 32$, code 6 bits zero + 10 bits (Δ_k)

where $\Delta_k = A_k - A_{k-1}$

A_k = numerical address of current block

A_{k-1} = numerical address of most recently encoded block.

The color quadrature components are encoded using dedicated Huffman code tables in coder unit 22. The tables listed as table number 2 and table number 3 in appendix A are used for the Q and I component values, respectively. After the block address and the color quadrature components have been encoded in the manner noted above, the first coefficient in diagonal memory unit 21 corresponding to the block, and which represents the average luminance of the block, is encoded using dedicated Huffman code table number 7 shown in appendix A. Thereafter, the cosine coefficients are processed for encoding by passing each cosine coefficient through a quantizer unit 23 in which the individual coefficients are divided by a distortion constant D_k supplied by distortion calculator unit 18. More particularly, each coefficient is multiplied by the quantity $1/D_k$ using a digital multiplier, and the resulting rounded product, designated as a quantized cosine coefficient, is coupled to

coder unit 22. In the preferred embodiment, each quantized cosine coefficient comprises a 12 bit digital character having 1 sign bit and 11 bits of magnitude. For each quantized cosine coefficient received in coder unit 22, the predictive mean value is calculated using the following relationship:

$$PM_K = \frac{1}{4} C_K + \frac{3}{4} PM_{K-1}$$

where PM_K is the predictive mean value of the K^{th} quantized coefficient, C_K is the value of the K^{th} quantized coefficient and PM_{K-1} is the predictive mean value of the $K-1^{th}$ quantized coefficient. The predictive mean value PM_K is used to select one of six of the Huffman code tables 1-6 listed in Appendix A to be used to encode the next appearing quantized coefficient, in the manner described below. Thus, PM_K is used to select the Huffman code table for quantized coefficient K+1, PM_{K+1} is used for quantized coefficient K+2, etc.

The four most significant magnitude bits of each quantized coefficient are next examined in coder unit 22 using conventional logic circuitry and, if the most significant four bits are zero, the quantized coefficient is Huffman coded using one of tables 1-6 listed in appendix A. Each table 1-6 is constructed using a different one of six probability distribution curves illustrated in Fig. 6. Each curve comprises an exponential function, with different curves having different mean values ranging from 1 to 32. The calculated predictive mean value PM_K measures the steepness of the probability curve for a given quantized coefficient, and thus each Huffman code table is selected for a particular quantized coefficient by converting the value of PM_K to the log (base 2) equivalent value, and using the converted value to specify the appropriate table. The table selected is ideally that table capable of encoding the quantized coefficient with the least number of bits.

When the four most significant bits of a quantized coefficient are non-zero, a Huffman coded special escape

symbol from the appropriate table and the actual twelve-bit quantized coefficient are transmitted. The escape symbol is the last symbol found in the Appendix A tables.

Coding of the transform coefficients proceeds as described until the predictive mean falls below a preselected fixed threshold, termed the run length threshold. When this occurs, a run length code corresponding to the number of successive quantized coefficients having value zero is generated by coder unit 22 using table number 8 from appendix A. If the zero run extends to the end of the block, a special end of block code is generated by coder unit 22 from table number 8. The above described encoding process is graphically illustrated in Fig. 7 in which the trend of the predictive mean values is illustrated by the solid curve labelled PM_i. The run length threshold is designated by the horizontal broken line, and the run length and end of block segments are designated with the legends RL and EOB, respectively. In the preferred embodiment, the numerical value of the run length threshold is one.

The code characters generated in coder unit 22 are stored in their order of generation in a rate buffer unit 25 having a predetermined maximum capacity N. Rate buffer 25 is a conventional unit capable of accepting binary input bits at a variable rate and generating bits at the output thereof at a constant rate of 2.39×10^5 bits/sec. in the preferred embodiment. Since the rate at which coder unit 22 supplies binary bits to the input of rate buffer 25 can vary widely, while the buffer output bit rate is constant, a rate feedback technique is incorporated into the encoder of Fig. 1 to minimize the probability of buffer overflow. For this purpose, a signal representative of the number of bits actually transferred from coder unit 22 to buffer unit 25 is coupled to distortion calculator unit 18 for each replenishment block K, and the value of the distortion constant D_k, which establishes the magnitude of the minimum quantization interval for quantizer unit 23, is recalculated. The calculation is performed in accordance with the following relationship:

DISTORTION CALCULATION

$$D = D'_K + K_D \cdot BFN(B_K - N/2)$$

where:

$$BFN(X) = \frac{X}{N - |X|}$$

D_K = Distortion parameter for block K

D'_K = Filtered distortion parameter

$$D'_K = T \cdot D'_{K-1} + (1-T)D_{K-1}$$

where

T = a constant (close to 1)

K_D = a constant

B_K = # of bits in buffer for block K

N = Max. number of bits

In addition, the distortion calculator 18 updates the value of the block difference threshold T_K for each encoded replenishment block in accordance with the following relationship:

REPLENISHMENT CALCULATION

$$T_K = T_{INIT} \quad \text{for } B_{LOW} \leq B_K \leq B_{HIGH}$$

$$T_K = T_{INIT} + K_R \cdot BFN(B_{LOW} - B_K) \quad \text{for } B_K < B_{LOW}$$

$$T_L = T_{INIT} + K_R \cdot BFN(B_K - B_{HIGH}) \quad \text{for } B_K > B_{HIGH}$$

where -

T_K = replenishment threshold for block K

T_{INIT} = initial threshold (about 5 for 8-bit input data)

K_R = multiplier constant (about 25-75)

B_{LOW} = low cutoff (about .1 of buffer)

B_{HIGH} = high cutoff (about .75 of buffer)

Thus, as the buffer unit 25 fullness increases, the quantizer unit 23 provides coarser quantization intervals for the encoding of the transform coefficients, which tends to reduce the number of bits per symbol generated by coder unit 22, and thus tends to reduce the buffer fullness. In addition, the block difference threshold T_k is raised, which tends to reduce the number of blocks selected for replenishment transformation, which also tends to reduce the buffer fullness. Similarly, when the buffer fullness decreases, the distortion constant D_k provides finer quantization intervals for processing the transform coefficients, which tends to increase the number of bits per symbol generated by coder unit 22; and the block difference threshold T_k is lowered, tending to select more blocks for replenishment processing, both of which tend to increase the buffer fullness.

The manner in which the serially generated code symbols representing the block replenishment information are arranged for transmission from buffer unit 25 to a decoder site is shown in Fig. 8, which illustrates one entire frame of information. As seen in this Fig., a frame of information commences with a frame sync code signal indicating the beginning of the frame, followed by a first control code signal B_K which specifies the state of buffer fullness at the beginning of the frame. This control code signal is followed by the second control code signal D_k, which is the

actual value of the distortion constant at the beginning of the frame. Following this header information, which is used to reset the decoder shown in Fig. 2 at the beginning of each frame, are groups of block symbols containing the block replenishment information. After the last such group, a new frame sync code signal indicates the beginning of the succeeding frame of information.

Each group of block replenishment code symbols commences with the block address code, is followed by the two color quadrature component code symbols and continues with the coefficient code symbols, as indicated in Fig. 8. The arrangement of the coefficient codes indicated in Fig. 8 for the first block to be updated corresponds to the representative plot shown in Fig. 7.

The block replenishment symbols encoded in the manner described above are transmitted over a suitable communication link to the decoder system shown in Fig. 2, which provides inverse processing for the received information code symbols. Thus, after receipt of a frame sync code signal, the initial value of the distortion constant D_k is coupled to an inverse quantizer unit 23' and the first block of replenishment information is initially decoded in decoder unit 22'. Decoder unit 22' contains the inverse code tables illustrated in appendix B which generate digital values from the received code symbols applied to the input thereof. The tables are arranged in a manner similar to that employed in coder unit 22, so that the block address codes, the color quadrature component codes and the DC coefficient code are all applied to their respective dedicated tables, while the cosine coefficient codes are applied on an individual basis to a selected one of six tables, depending on the value of the predictive mean calculated for each received quantized coefficient code. The emerging twelve-bit digital characters representing the quantized coefficients are inverse quantized in unit 23' by simply digitally multiplying each twelve-bit character with the value of D_k, and the resulting inversely quantized cosine coefficients are stored in diagonal memory unit 21', along

with the corresponding block address digital character, the digital character representing the DC term and the quadrature chrominance characters. The coefficients are transformed to the time domain by subjecting the coefficients to a two-step inverse cosine transformation in unit 20', with the intermediate resulting values being stored back in diagonal memory unit 21'. After the inverse cosine transformation process has been completed, the resulting pixel samples are stored in output memory unit 13, replacing the former eight by eight block of pixel information. It should be noted that the field of information stored in output memory unit 13' comprises a replica of the field of information stored in reference memory unit 14 of the Fig. 1 encoder.

The replenished field information contained in output memory unit 13' is coupled to an output digital processor unit 12', and thence to analog processor unit 11' which converts the digital video data to analog form. The emerging analog video signals are coupled to a suitable utilization device, such as a raster scan monitor.

As binary bits are transferred from buffer unit 25' to decoder unit 22', a signal representing the number of bits transferred is coupled to distortion calculator unit 18', which updates the value of the distortion constant D_k for each block of replenishment information. The distortion calculator 18' employs the same algorithm as that noted above for distortion calculator 18.

Further compression of the input video is obtained by two additional signal processing techniques. Firstly, the luminance portion of a field of input video is subsampled to provide 256 lines/frame and 256 samples/line (65,536 pixels/frame), as opposed to the normal standard of 512 lines/frame and 512 samples/line (262,144 pixels/frame). After decoding, the full luminance field is recovered in output digital processor 12' by interpolative processing of the actual luminance samples transmitted through the system. Deleted luminance samples are reconstructed by summing adjacent samples and dividing the result by two. Thus, if A

and B are adjacent luminance samples in a line the intermediate sample is reconstructed by forming the sum $\frac{A}{2} + \frac{B}{2}$, and inserting this result between sample A and sample B.

The quadrature chrominance components for each 8x8 block of input video are compressed by first discarding the 2 least significant bits of each 8-bit digital chrominance quadrature component sample to form 6-bit digital characters. For each 8x8 block, the separate quadrature component samples are averaged over the block by summing every other component sample in a given line, dividing the result by four, storing the result obtained for each line, summing the result for the eight lines in a block and dividing the result by eight. Specifically, for the I quadrature component for block K, the initial sample array is:

$$
\begin{array}{ccccccc}
I_{11} & I_{12} & I_{13} & I_{14} & I_{16} & I_{17} & I_{18} \\
I_{21} & - & - & - & - & - & I_{28} \\
| & & & & & & | \\
| & & & & & & | \\
| & & & & & & | \\
| & & & & & & | \\
I_{81} & - & - & - & - & - & I_{88}
\end{array}
$$

The first row average is:

$$I_{AV1} = (I_{11} + I_{13} + I_{15} + I_{17})/4$$

The last row average is:

$$I_{AV8} = (I_{81} + I_{83} + I_{85} + I_{87})/4$$

The block average is:

$$I_{AV\ BLOCK\ K} = (I_{AV1} + I_{AV2} + \ldots + I_{AV8})/8$$

For the Q quadrature component, the first row average is:

$$Q_{AV1} = (Q_{12} + Q_{14} + Q_{16} + Q_{18})/4$$

The last row average is:

$$Q_{AV8} = (Q_{82} + Q_{84} + Q_{86} + Q_{88})/4$$

The block average is:

$$Q_{AV\ BLOCK\ K} = (A_{AV1} + Q_{AV2} + \dots + Q_{AV8})/8$$

The resulting block average for each component is stored as a 6-bit character in input memory 13. The full chrominance field is recovered in output digital processor 12' by inverse interpolative processing of the average chrominance samples transmitted through the system.

While suitable for many applications requiring information signal compression, the invention is especially adapted for use in a video teleconferencing system in which the prime criterion is bandwith reduction with minimum degradation in the subjective quality of the video images. At a typical sampling rate of 10.7 MHz for digital video transmission using eight-bits per digital sample character, the required bit rate to reliably transmit video information without compression is 8.56×10^7 bits per second. By processing video signals according to the invention, using the same sample rate and the same size digital characters (i.e. eight-bits) in the analog to digital converter section of the encoder, and the digital to analog section of the decoder, compressed digital video can be transmitted from the encoder buffer unit 25 to the decoder buffer unit 25' at a rate of 2.39×10^5 bits per second, which is .25 percent of the standard uncompressed digital bit rate. As will be appreciated by those skilled in the art, such a substantial reduction in the bit rate enables video information of good picture quality to be transmitted over a communication link having a substantially narrower bandwith, for example four conventional digital voice channels, with the result that substantially more information traffic can be routed over available communication links.

While the above provides a full and complete disclosure of the preferred embodiments of the invention, various modifications, alternate constructions and equiva-

lents may be employed without departing from the true spirit and scope of the invention. For example, while eight by eight pixel blocks are employed in the preferred embodiment of the invention, blocks of other sizes may be employed, if desired. The relevant criteria for selecting appropriate block sizes are the processing time required by the block difference and decision unit 16, the cosine transform unit 20, the quantizer unit 23, the coder unit 22 and the distortion calculator 18. In general, larger blocks require more processing time, and the speed of currently available digital circuitry provides a practical limitation of about 32 pixels by 32 pixels on the maximum block size. In addition, for applications in which the amount of interframe image motion is excessive (i.e. greater than that normally present in video conferencing applications), a smaller block size may be necessary in order to provide decoded video signals of good subjective quality. Selection of smaller block sizes, however, increases the required minimum bit rate for the buffer units 25, 25'. In addition, different weighting factors may be employed for field merging, if desired; however, in the development of the preferred embodiment it has been discovered that a ratio of seven to one results in decoded video signals which are quite blurry, while a ratio approaching one to one results in a substantially increased number of blocks selected for replenishment, requiring a higher minimum bit rate for reliable transmission and decoding. The above description and illustrations, therefore, should not be construed as limiting the scope of the invention, which is defined by the appended claims.

Ø.	16	99DB
Ø.	16	99DA
Ø.	16	99D9
Ø.	16	99D8
Ø.	16	99D7
Ø.	16	99D6
Ø.	16	99D5
Ø.	16	99D4
Ø.	16	99D3
Ø.	16	99D2
Ø.	16	99D1
Ø.	16	99DØ
Ø.	16	99CF
Ø.	16	99CE
Ø.	16	99CD
Ø.	16	99CC
Ø.	16	99CB
Ø.	16	99CA
Ø.	16	99C9
Ø.	16	99C8
Ø.	16	99C7
Ø.	16	99C6
Ø.	16	99C5
Ø.	16	99C4
Ø.	16	99C3
Ø.	16	99C2
Ø.	16	99C1
Ø.	16	99CØ
Ø.	16	99BF
Ø.	16	99BE
Ø.	16	99BD
Ø.	16	99BC
Ø.	16	99BB
Ø.	16	99BA
Ø.	16	99B9
Ø.	16	99B8
Ø.	16	99B7
Ø.	16	99B6
Ø.	16	99B5
Ø.	16	99B4
Ø.	16	99B3
Ø.	16	99B2
Ø.	16	99B1
Ø.	16	99BØ
Ø.	16	99AF
Ø.	16	99AE
Ø.	16	99AD
Ø.	16	99AC
Ø.	16	99AB
Ø.	16	99AA
Ø.	16	99A9
Ø.	16	99A8
Ø.	16	99A7
Ø.	16	99A6
Ø.	16	99A5
Ø.	16	99A4
Ø.	16	99A3
Ø.	16	99A2
Ø.	16	99A1
Ø.	16	99AØ

Symbols = 142561.
l Entropy = 1.5Ø7
'er Symbol = 1.7Ø6

TABLE 2

Entry	Occurrences	Length	Huffman Code
0	19459.	1	0000
1	13767.	2	0003
2	4937.	3	0005
3	2096.	4	0008
4	1016.	6	0027
5	568.	6	0025
6	352.	7	004C
7	214.	7	0048
8	136.	8	0093
9	92.	9	0136
10	52.	10	026F
11	42.	10	026A
12	29.	10	0249
13	32.	10	024A
14	19.	11	04D2
15	19.	11	04D1
16	25.	11	04DC
17	9.	12	09A0
18	13.	11	0491
19	13.	11	0490
20	6.	13	1377
21	1.	15	4DD7
22	4.	13	125F
23	2.	14	26BC
24	4.	13	125E
25	2.	14	26BB
26	3.	13	125B
27	0.	15	4970
28	6.	13	1376
29	3.	13	125A
30	4.	13	125D
31	3.	13	1259
32	3.	13	1258
33	1.	15	4DD6
34	0.	16	9AD7
35	1.	15	4DD5
36	1.	15	4DD4
37	0.	16	9AD6
38	1.	15	4DD3
39	0.	16	9AD5
40	0.	16	9AD4
41	0.	16	9AD3
42	0.	16	9AD2
43	1.	15	4DD2
44	2.	14	26BA
45	1.	15	4DD1
46	1.	15	4DD0
47	1.	15	4D7F
48	2.	14	26B9
49	1.	15	4D7E
50	1.	15	4D7D
51	2.	14	26B8
52	0.	16	9AD1
53	0.	16	9AD0
54	0.	16	9ACF
55	0.	16	9ACE
56	0.	16	9ACD
57	1.	15	4D7C
58	0.	16	9ACC
59	0.	16	9ACB
60	0.	16	9ACA
61	0.	16	9AC9
62	0.	16	9AC8
63	1.	15	4D7B
64	0.	16	9AC7
65	1.	15	4D7A
66	0.	16	9AC6
67	0.	16	9AC5
68	2.	14	26B7

Ø.	16	9AC4
Ø.	16	9AC3
2.	14	26B6
Ø.	16	9AC2
Ø.	16	9AC1
Ø.	16	9ACØ
Ø.	16	9A7F
Ø.	16	9A7E
Ø.	16	9A7D
Ø.	16	9A7C
Ø.	16	9A7B
Ø.	16	9A7A
Ø.	16	9A79
Ø.	16	9A78
Ø.	16	9A77
Ø.	16	9A76
Ø.	16	9A75
Ø.	16	9A74
Ø.	16	9A73
Ø.	16	9A72
Ø.	16	9A71
Ø.	16	9A7Ø
Ø.	16	9A6F
Ø.	16	9A6E
Ø.	16	9A6D
Ø.	16	9A6C
Ø.	16	9A6B
Ø.	16	9A6A
Ø.	16	9A69
Ø.	16	9A68
Ø.	16	9A67
Ø.	16	9A66
Ø.	16	9A65
Ø.	16	9A64
Ø.	16	9A63
Ø.	16	9A62
Ø.	16	9A61
Ø.	16	9A6Ø
Ø.	16	9A1F
Ø.	16	9A1E
Ø.	16	9A1D
Ø.	16	9A1C
Ø.	16	9A1B
Ø.	16	9A1A
Ø.	16	9A19
Ø.	16	9A18
Ø.	16	9A17
Ø.	16	9A16
Ø.	16	9A15
Ø.	16	9A14
Ø.	16	9A13
Ø.	16	9A12
Ø.	16	9A11
Ø.	16	9A1Ø
Ø.	16	92E7
Ø.	16	92E6
Ø.	16	92E5
Ø.	16	92E4
Ø.	16	92E3
Ø.	16	92E2

ymbols = 42954.
Entropy = 2.Ø48
r Symbol = 2.Ø74

TABLE 3

ntry	Occurrences	Length	Huffman Code
0	16170.	2	0003
1	14015.	2	0002
2	6726.	3	0003
3	3891.	3	0001
4	2383.	4	0004
5	1615.	4	0000
6	1161.	5	0003
7	722.	6	0016
8	545.	6	0014
9	414.	7	002F
10	350.	7	002B
11	259.	7	000B
12	194.	8	005D
13	141.	8	0054
14	115.	8	0013
15	101.	8	0011
16	62.	9	002A
17	72.	9	00AA
18	46.	9	0020
19	56.	9	0025
20	41.	10	0170
21	29.	10	0050
22	31.	10	0053
23	20.	11	02E2
24	33.	10	0057
25	23.	11	02E7
26	21.	11	02E4
27	19.	11	02AE
28	15.	11	00A5
29	21.	11	02E3
30	13.	11	0090
31	16.	11	00AC
32	14.	11	0093
33	8.	12	015B
34	4.	13	0AB3
35	7.	12	0148
36	7.	12	0147
37	9.	12	055A
38	6.	12	010F
39	6.	12	010E
40	7.	12	0146
41	5.	13	0B96
42	5.	13	0B95
43	8.	12	015A
44	4.	13	0AB2
45	7.	12	0145
46	4.	13	0AB1
47	5.	13	0B94
48	3.	13	0248
49	6.	12	010D
50	7.	12	0144
51	4.	13	0AB0
52	3.	13	024A
53	1.	14	0423
54	6.	12	010C
55	0.	16	1495
56	2.	14	157D
57	1.	14	0422
58	3.	13	0249
59	2.	14	157C
60	1.	14	0421
61	3.	13	024B
62	1.	14	0420
63	3.	13	0247
64	3.	13	0246
65	2.	14	157B
66	2.	14	157A
67	3.	13	0245
68	1.	15	2E6F

69	2.	14	1579
7Ø	1.	15	2E6E
71	1.	15	2E6D
72	2.	14	1578
73	4.	13	Ø293
74	5.	13	ØABF
75	1.	15	2E6C
76	3.	13	Ø244
77	6.	12	Ø1ØB
78	1.	15	2E6B
79	Ø.	16	1494
8Ø	1.	15	2E6A
81	1.	15	2E69
82	1.	15	2E68
83	1.	15	2E67
84	1.	15	2E66
85	Ø.	16	1493
86	1.	15	2E65
87	3.	13	Ø215
88	Ø.	16	1492
89	1.	15	2E64
9Ø	1.	15	[illegible]63
91	1.	15	2E62
92	2.	14	156F
93	1.	15	2E61
94	Ø.	16	1491
95	2.	14	156E
96	1.	15	2E6Ø
97	Ø.	16	149Ø
98	Ø.	16	1ØA7
99	Ø.	16	1ØA6
1ØØ	Ø.	16	1ØA5
1Ø1	1.	15	2E5F
1Ø2	Ø.	16	1ØA4
1Ø3	1.	15	2E5E
1Ø4	2.	14	156D
1Ø5	1.	15	2E5D
1Ø6	Ø.	16	1ØA3
1Ø7	Ø.	16	1ØA2
1Ø8	Ø.	16	1ØA1
1Ø9	Ø.	16	1ØAØ
11Ø	Ø.	16	1Ø9F
111	Ø.	16	1Ø9E
112	Ø.	16	1Ø9D
113	1.	15	2E5C
114	2.	14	156C
115	Ø.	16	1Ø9C
116	Ø.	16	1Ø9B
117	Ø.	16	1Ø9A
118	Ø.	16	1Ø99
119	Ø.	16	1Ø98
12Ø	Ø.	16	1Ø97
121	Ø.	16	1Ø96
122	Ø.	16	1Ø95
123	Ø.	16	1Ø94
124	Ø.	16	1Ø93
125	1.	15	ØA4B
126	Ø.	16	1Ø92
127	Ø.	16	1Ø91
128	Ø.	16	1Ø9Ø

otal Symbols = 58336.
ymbol Entropy = 2.774
its Per Symbol = 2.834

****

TABLE 4

Entry *	Occurrences *	Length *	Huffman Code ***
0	5652.	3	0007
1	6440.	2	0001
2	3916.	3	0005
3	2656.	3	0000
4	2019.	4	0009
5	1581.	4	0003
6	1229.	5	001A
7	979.	5	0018
8	811.	5	0005
9	618.	6	0036
10	516.	6	0032
11	417.	6	0021
12	404.	6	0020
13	321.	7	006E
14	256.	7	0047
15	211.	7	0044
16	181.	7	0011
17	177.	8	00DF
18	156.	8	00CF
19	140.	8	00CD
20	103.	8	0027
21	93.	8	0024
22	98.	8	0026
23	88.	8	0020
24	71.	9	019C
25	68.	9	0196
26	63.	9	0119
27	59.	9	0117
28	50.	9	004B
29	44.	9	0042
30	39.	10	0379
31	35.	10	033A
32	32.	10	0235
33	31.	10	0234
34	33.	10	0236
35	27.	10	022A
36	24.	10	0095
37	22.	10	0086
38	28.	10	022D
39	21.	11	06F6
40	23.	10	0094
41	19.	11	06F0
42	17.	11	0666
43	17.	11	0665
44	9.	12	0CED
45	17.	11	0664
46	15.	11	0462
47	10.	12	0DEA
48	8.	12	08DE
49	12.	11	0451
50	16.	11	046E
51	10.	12	0DE9
52	10.	12	0DE8
53	11.	11	010E
54	12.	11	0450
55	7.	12	08C6
56	10.	12	0DE3
57	6.	12	08A6
58	2.	14	378B
59	5.	13	1BDC
60	4.	13	199F
61	7.	12	08C3
62	3.	13	115B
63	4.	13	199E
64	7.	12	08C2
65	4.	13	199D
66	7.	12	08C1
67	5.	13	1BD7
68	2.	14	370A

69	3.	13	115A
70	7.	12	08C0
71	8.	12	08C7
72	4.	13	199C
73	3.	13	1159
74	7.	12	08AF
75	5.	13	1BD6
76	6.	12	08A5
77	2.	14	3789
78	2.	14	3788
79	1.	14	087A
80	3.	13	1158
81	6.	12	08A4
82	2.	14	33BF
83	7.	12	0BAE
84	2.	14	33BE
85	2.	14	33BD
86	4.	13	11BF
87	2.	14	33BC
88	3.	13	114F
89	2.	14	33BB
90	0.	16	CEC1
91	2.	14	33BA
92	3.	13	114E
93	3.	13	043F
94	2.	14	33B9
95	1.	14	0879
96	1.	14	0878
97	1.	15	6F7F
98	0.	16	CEC0
99	0.	16	21F7
100	1.	15	6F7E
101	1.	15	6F7D
102	2.	14	33B8
103	0.	16	21F6
104	2.	14	33B3
105	1.	15	6F7C
106	4.	13	11BE
107	1.	15	6F7B
108	0.	16	21F5
109	0.	16	21F4
110	0.	16	21F3
111	1.	15	6F7A
112	0.	16	21F2
113	0.	16	21F1
114	2.	14	33B2
115	1.	15	6F79
116	1.	15	6F78
117	0.	16	21F0
118	0.	16	21EF
119	0.	16	21EE
120	1.	15	6F77
121	0.	16	21ED
122	0.	16	21EC
123	2.	14	33B1
124	1.	15	6F76
125	1.	15	6F75
126	1.	15	6F74
127	1.	15	6761
128	28.	10	022C

* Total Symb s = 30144.
* Symbol Ent py = 3.793
* Bits Per S bol = 3.835

TABLE 5

Entry	Occurrences	Length	Huffman Code
Ø	166Ø.	3	ØØØ4
1	1ØØ4.	3	ØØØ5
2	1425.	3	ØØØ2
3	978.	4	ØØØD
4	822.	4	ØØØ7
5	637.	4	ØØØ2
6	595.	4	ØØØØ
7	522.	5	ØØ1E
8	411.	5	ØØ18
9	393.	5	ØØØC
1Ø	372.	5	ØØØ7
11	3Ø8.	5	ØØØ2
12	27Ø.	6	ØØ3E
13	248.	6	ØØ3A
14	234.	6	ØØ33
15	236.	6	ØØ38
16	161.	6	ØØØC
17	143.	7	ØØ7E
18	123.	7	ØØ73
19	131.	7	ØØ77
2Ø	94.	7	ØØ1B
21	1Ø2.	7	ØØ36
22	96.	7	ØØ34
23	76.	7	ØØØD
24	61.	8	ØØE5
25	76.	7	ØØØC
26	53.	8	ØØ6F
27	56.	8	ØØCA
28	57.	8	ØØCB
29	4Ø.	8	ØØ1E
3Ø	31.	9	Ø1DA
31	43.	8	ØØ34
32	28.	9	Ø193
33	31.	9	Ø1D9
34	39.	8	ØØ1C
35	25.	9	ØØDC
36	37.	9	Ø1FE
37	27.	9	Ø192
38	19.	9	ØØ3A
39	17.	1Ø	Ø3F9
4Ø	29.	9	Ø1C8
41	23.	9	ØØD4
42	11.	1Ø	ØØD7
43	2Ø.	9	ØØ3B
44	16.	1Ø	Ø3B6
45	13.	1Ø	Ø32Ø
46	18.	1Ø	Ø3FB
47	27.	9	Ø191
48	16.	1Ø	Ø393
49	6.	11	Ø35E
5Ø	12.	1Ø	Ø1AC
51	11.	1Ø	ØØD6
52	11.	1Ø	ØØD5
53	5.	11	Ø1A8
54	12.	1Ø	Ø1AB
55	9.	11	Ø7FE
56	6.	11	Ø35D
57	9.	11	Ø7FD
58	12.	1Ø	Ø1AA
59	4.	12	ØFE8
6Ø	6.	11	Ø35C
61	5.	11	ØØFF
62	7.	11	Ø725
63	17.	1Ø	Ø3F8
64	7.	11	Ø724
65	1Ø.	1Ø	ØØ7D
66	5.	11	ØØFE
67	4.	12	ØEDF
68	5.	11	ØØFD

59	5.	11	ØØFC
7Ø	1.	13	Ø6A5
71	2.	13	1FF2
72	3.	12	ØCB4
73	4.	12	ØEDE
74	6.	11	Ø35B
75	3.	12	Ø6EF
76	7.	11	Ø643
77	3.	12	Ø6EE
78	4.	12	ØEDD
79	5.	11	ØØF9
8Ø	3.	12	Ø6ED
81	Ø.	15	6429
82	1.	13	Ø6A4
83	3.	12	Ø6EC
84	6.	11	Ø35A
85	3.	12	Ø6EB
86	3.	12	Ø6EA
87	2.	3	1FF1
88	3.		Ø6E9
89	1.		Ø3E3
9Ø	1.		Ø3E2
91	Ø.		6428
92	Ø.	6	C85F
93	2.	13	1FFØ
94	2.	13	1FD7
95	Ø.	16	C85E
96	1.	13	Ø3E1
97	Ø.	16	C85D
98	1.	13	Ø3EØ
99	2.	13	1FD6
ØØ	2.	13	1FD5
Ø1	2.	13	1FD4
Ø2	3.	12	Ø6E8
Ø3	Ø.	16	C85C
Ø4	Ø.	16	C85B
Ø5	1.	14	3FFF
Ø6	1.	14	3FFE
Ø7	2.	13	1FD3
Ø8	4.	12	ØEDC
Ø9	2.	13	1FD2
1Ø	1.	14	3FFD
11	Ø.	16	C85A
12	3.	12	Ø6BF
13	1.	14	3FFC
14	1.	14	3FFB
15	1.	14	3FFA
16	1.	14	3FF9
17	Ø.	16	C859
18	3.	12	Ø6BE
19	Ø.	16	C858
2Ø	1.	14	3FF8
21	Ø.	16	C857
22	1.	14	3FE7
23	Ø.	16	C856
24	Ø.	16	C855
25	1.	14	3FE6
26	3.	12	Ø353
27	Ø.	16	C854
28	31.	9	Ø1D8

ta bols = 12959.
mt ntropy = 4.557
ta r Symbol = 4.578

TABLE 6

Entry	Occurrences	Length	Huffman Code
0	496.	3	0005
1	366.	3	0001
2	294.	4	000E
3	232.	4	000B
4	213.	4	0005
5	157.	5	001F
6	142.	5	001B
7	134.	5	0019
8	118.	5	0012
9	104.	5	0009
10	88.	5	0001
11	96.	5	0003
12	67.	6	0031
13	71.	6	0035
14	54.	6	001A
15	57.	6	001E
16	60.	6	001F
17	55.	6	001C
18	66.	6	0027
19	45.	6	0004
20	44.	6	0001
21	54.	6	0019
22	39.	7	007A
23	27.	7	0037
24	35.	7	0060
25	31.	7	004D
26	27.	7	0036
27	36.	7	007B
28	26.	7	0023
29	25.	7	0022
30	21.	8	00F7
31	24.	7	000B
32	28.	7	003B
33	22.	7	0001
34	25.	7	0021
35	19.	8	00F3
36	16.	8	00C0
37	13.	8	0063
38	14.	8	0098
39	13.	8	0062
40	14.	8	0075
41	13.	8	0061
42	11.	8	0015
43	13.	8	0060
44	9.	9	01E4
45	11.	8	0014
46	10.	9	01ED
47	10.	9	01EC
48	8.	9	0133
49	11.	8	0001
50	6.	9	00B3
51	9.	9	01A7
52	5.	9	0000
53	5.	10	03CB
54	4.	10	0309
55	4.	10	0308
56	4.	10	0307
57	4.	10	0306
58	2.	11	0695
59	1.	11	0005
60	4.	10	03C5
61	7.	9	00E9
62	3.	10	01D1
63	4.	10	0304
64	4.	10	0265
65	2.	11	0694
66	2.	11	0693
67	2.	11	0692
68	2.	11	0691

69	2.	11	0690
70	6.	9	0002
71	0.	14	2645
72	2.	11	061F
73	1.	11	0004
74	2.	11	061E
75	2.	11	061D
76	2.	11	061C
77	1.	12	0F2B
78	2.	11	061B
79	3.	10	01D0
80	0.	14	2644
81	0.	14	2643
82	2.	11	061A
83	0.	14	2642
84	0.	14	2641
85	1.	12	0F2A
86	0.	14	2640
87	0.	15	4C9F
88	1.	12	0F29
89	1.	12	0F28
90	2.	11	0619
91	0.	15	4C9E
92	2.	11	0618
93	1.	12	0D37
94	0.	15	4C9D
95	2.	11	0617
96	3.	10	0003
97	1.	12	0D36
98	1.	12	0D35
99	1.	12	0D34
100	1.	12	0D33
101	0.	15	4C9C
102	0.	15	4C9B
103	0.	15	4C9A
104	0.	15	4C99
105	1.	12	0D32
106	0.	15	4C98
107	0.	15	4C97
108	0.	15	4C96
109	0.	15	4C95
110	1.	12	0D31
111	1.	12	0D30
112	0.	15	4C94
113	2.	11	0616
114	0.	15	4C93
115	0.	15	4C92
116	1.	12	0D2F
117	1.	12	0D2E
118	0.	15	4C91
119	1.	12	0D2D
120	0.	15	4C90
121	0.	15	4C8F
122	1.	12	0D2C
123	0.	15	4C8E
124	0.	15	4C8D
125	2.	11	0615
126	0.	15	4C8C
127	2.	11	0614
128	12.	8	0040

' Total Symbols = 3705.
' Symbol Entropy = 5.023
' Bits Per Symbol = 5.047

TABLE 7

Entry	Occurrences	Length	Huffman Code
0	1134.	4	000D
1	476.	5	0011
2	463.	5	000F
3	482.	5	0012
4	607.	5	001C
5	632.	5	001F
6	611.	5	001D
7	546.	5	001B
8	489.	5	0014
9	500.	5	0015
10	503.	5	0016
11	423.	5	000C
12	317.	6	003D
13	353.	5	0005
14	344.	5	0004
15	381.	5	0007
16	241.	6	0026
17	197.	6	0012
18	266.	6	002F
19	233.	6	0021
20	220.	6	001C
21	170.	6	0007
22	149.	7	0067
23	213.	6	001A
24	243.	6	0027
25	206.	6	0015
26	216.	6	001B
27	162.	6	0002
28	230.	6	0023
29	200.	6	0013
30	193.	6	0011
31	181.	6	000C
32	160.	6	0001
33	278.	6	0032
34	209.	6	0016
35	191.	6	0010
36	154.	7	0078
37	168.	6	0005
38	109.	7	003A
39	170.	6	0006
40	159.	6	0000
41	101.	7	0029
42	100.	7	0028
43	105.	7	002E
44	77.	8	00F2
45	56.	8	0076
46	62.	8	0077
47	79.	8	00F3
48	81.	7	0008
49	94.	7	001B
50	89.	7	001A
51	125.	7	005C
52	165.	6	0003
53	149.	7	0056
54	55.	8	005F
55	22.	9	0026
56	21.	9	0025
57	28.	9	0174
58	23.	9	0027
59	17.	10	02EB
60	17.	10	02EA
61	13.	10	017A
62	13.	10	0179
63	14.	10	017B
64	1.	13	00C4
65	0.	15	2F14
66	1.	13	09C3
67	0.	15	091C
68	0.	15	091A

69	1.	13	0BC2
70	0.	15	0919
71	0.	15	0918
72	1.	13	0BC1
73	0.	15	0917
74	0.	15	0916
75	0.	15	0915
76	1.	13	0BC0
77	1.	13	024F
78	0.	15	0914
79	0.	15	0913
80	1.	13	024E
81	0.	15	0912
82	0.	15	0911
83	0.	15	0910
84	0.	15	090F
85	0.	15	090E
86	2.	12	0124
87	0.	15	090D
88	1.	13	024D
89	0.	15	090C
90	1.	13	024C
91	0.	15	090B
92	0.	15	090A
93	0.	15	0909
94	0.	15	0908
95	0.	15	0907
96	1.	13	024B
97	0.	15	0906
98	0.	15	0905
99	0.	15	0904
100	0.	15	0903
101	1.	13	024A
102	0.	15	0902
103	0.	15	0901
104	0.	15	0900
105	1.	13	0247
106	0.	16	5E3F
107	0.	16	5E3E
108	0.	16	5E3D
109	0.	16	5E3C
110	0.	16	5E3B
111	0.	16	5E3A
112	0.	16	5E39
113	0.	16	5E38
114	0.	16	5E37
115	0.	16	5E36
116	0.	16	5E35
117	0.	16	5E34
118	0.	16	5E33
119	0.	16	5E32
120	0.	16	5E31
121	0.	16	5E30
122	0.	16	5E2F
123	0.	16	5E2E
124	0.	16	5E2D
125	0.	16	5E2C
126	0.	16	5E2B
127	0.	16	5E2A
128	69.	8	00BB

* Total Symbols = 14768.
* Symbol Entropy = 5.538
* Bits Per Symbol = 5.570

TABLE 8

Entry	Occurrences	Length	Huffman Code
0	12200.	2	0001
1	6198.	3	0004
2	3859.	4	000B
3	2618.	4	0002
4	2085.	4	0000
5	1716.	5	0014
6	1378.	5	0003
7	1069.	6	002B
8	950.	6	002A
9	815.	6	000E
10	767.	6	000D
11	621.	6	0005
12	471.	7	001F
13	401.	7	0019
14	349.	7	001B
15	151.	8	0012
16	98.	9	007B
17	63.	9	0023
18	57.	9	0020
19	44.	10	004F
20	58.	9	0022
21	52.	10	00F3
22	58.	9	0021
23	55.	10	00F7
24	53.	10	00F5
25	55.	10	00F6
26	52.	10	00F2
27	34.	10	004C
28	14.	12	03D2
29	6.	13	07A3
30	10.	12	03D0
31	3.	13	026B
32	4.	13	026A
33	2.	14	04EB
34	7.	13	07A7
35	10.	12	013B
36	2.	14	04EA
37	3.	14	0F4D
38	3.	14	0F4C
39	1.	14	04D2
40	0.	16	13A7
41	1.	15	1E8B
42	1.	15	1E8A
43	1.	15	1E89
44	0.	16	13A6
45	0.	16	13A5
46	0.	16	13A4
47	1.	15	1E88
48	0.	16	13A3
49	0.	16	13A2
50	0.	16	13A1
51	0.	16	13A0
52	0.	16	139F
53	0.	16	139E
54	0.	16	139D
55	0.	16	139C
56	0.	16	139B
57	0.	16	139A
58	0.	16	1399
59	0.	16	1398
60	0.	16	1397
61	0.	16	1396
62	0.	16	1395
63	0.	16	1394
64	0.	16	1393
65	0.	16	1392
66	0.	16	1391
67	0.	16	1390
68	0.	16	138F

0.	16	138E
0.	16	138D
0.	16	138C
0.	16	138B
0.	16	138A
0.	16	1389
0.	16	1388
0.	16	1387
0.	16	1386
0.	16	1385
0.	16	1384
0.	16	1383
0.	16	1382
0.	16	1381
0.	16	1380
0.	16	137F
0.	16	137E
0.	16	137D
0.	16	137C
0.	16	137B
0.	16	137A
0.	16	1379
0.	16	1378
0.	16	1377
0.	16	1376
0.	16	1375
0.	16	1374
0.	16	1373
0.	16	1372
0.	16	1371
0.	16	1370
0.	16	136F
0.	16	136E
0.	16	136D
0.	16	136C
0.	16	136B
0.	16	136A
0.	16	1369
0.	16	1368
0.	16	1367
0.	16	1366
0.	16	1365
0.	16	1364
0.	16	1363
0.	16	1362
0.	16	1361
0.	16	1360
0.	16	135F
0.	16	135E
0.	16	135D
0.	16	135C
0.	16	135B
0.	16	135A
0.	16	1359
0.	16	1358
0.	16	134F
0.	16	134E
0.	16	134D
0.	16	134C
14000.	2	0003

Symbols = 58475.
Entropy = 3.216
er Symbol = 3.235

CLAIMS:

1. A method for processing time domain information signals having a successive field format to effect substantial compression of said signals, said method comprising the steps of:

comparing corresponding blocks of time domain information signals from successive fields;

converting a block of said time domain information signals to a transform domain signal represented by a D.C. coefficient representing the average intensity of a converted block and a plurality of discrete cosine transform coefficients when the difference between said corresponding blocks exceeds a first variable parametric value; and

encoding said transform domain coefficients for subsequent utilization.

2. A method of encoding transform coefficients representing time domain information signals having a successive field format prior to transmission over a communication link in order to effect substantial compression of said signals, said transform coefficients being arranged in a plurality of groups, each group representing an N by N block of field information signals, said method comprising the steps of:

(a) providing a plurality of code tables:

(b) generating a block address code from a first dedicated one of said code tables, said block address code representing the field address of the block represented by a group of said transform coefficients;

(c) generating a DC coefficient code representing the average intensity of said block from a second dedicated one of said plurality of code tables; and

(d) generating a succession of codes representing the remaining transform coefficients corresponding to said block by calculating the predictive value of each said remaining transform coefficient, selecting one of said plurality of code tables in accordance with said predictive value, and generating a code representing the corresponding transform coefficient from said selected table.

3. The method of claim 2 wherein said step of calculating is performed in accordance with the formula $PM_K=1/4\ C_K+3/4\ PM_{K-1}$, where PM_K is the predictive mean value of the K^{th} coefficient, C_K is the actual value of the K^{th}

coefficient and PM_{K-1} is the predictive mean value of the $K-1^{th}$ coefficient.

4. The method of claim 2 wherein said time domain information signals are color video signals having quadrature components, and wherein said method of encoding includes the steps of providing individual code tables for said quadrature components, calculating the average value of each quadrature component for said block, and selecting a code value representing said average value from the corresponding individual quadrature component table.

5. The method of claim 2 wherein said method of encoding further includes the steps of comparing each said predictive value with a preselected threshold value, and generating a zero run length code specifying the total number of successive predictive values lying below said preselected threshold value.

6. The method of claim 5 wherein said codes are multi-bit binary codes, and wherein said method further includes the steps of transferring said codes to a buffer in the order of generation, monitoring the number of bits transferred to said buffer, and varying said variable parametric value in accordance with the following formula:

$$D = D'_K + K_D \cdot BFN(B_K - N/2)$$

where

$$BFN(X) = \frac{X}{N-|X|}$$

D_K = Distortion parameter for block K

D'_K = Filtered distortion parameter

$$D'_K = T \cdot D'_{K-1} + (1-T)D_{K-1}$$

where

T = a constant (close to 1)

K_D = a constant

B_K = # of bits in buffer for block K

N = Max. number of bits

7. The method of claim 5 further including the step of generating an end of block code when the predictive values for successive remaining transform coefficients in said block lie below said preselected threshold value.

8. The method of claim 2 wherein said transform coefficients comprise discrete cosine transform coefficients.

9. A method for processing time domain information signals for transmission over a communication link, said time domain information signals having a successive field format, said method comprising the steps of:

(a) generating a frame sync code signal indicating the beginning of a frame;

(b) generating a first control code signal B_K representative of the fullness of a transmission rate buffer at the beginning of said frame;

(c) generating a second control code signal D_K representative of a first variable parametric value at the beginning of said frame; and

(d) generating a plurality of block replenishment code symbols each representative of the value of transform coefficients corresponding to individual sub-field blocks having interfield block differences greater than a second variable parametric value, each said block replenishment code symbol including a block address code specifying the field address of the corresponding block, a DC code term

representative of the average intensity of the corresponding block, and a plurality of coefficient code terms representative of the value of discrete cosine transform coefficients for said corresponding block.

10. The method of claim 9 wherein said time domain information signals are color video signals having quadrature components, and wherein said step (d) of generating includes the step of providing first and second color code terms in each of said plurality of said block replenishment code symbols representing the average value of each quadrature component for said corresponding block between said block address code and said DC code term.

11. The method of claim 9 wherein said step (d) of generating includes the step of providing a run length code term specifying the total number of successive transform coefficient zero values having a predictive mean value less than a preselected fixed threshold value.

12. The method of claim 9 wherein said step (d) of generating includes a step of providing an end of block code term when the predictive values for successive remaining transform coefficients in said corresponding block lie below a preselected fixed threshold value.

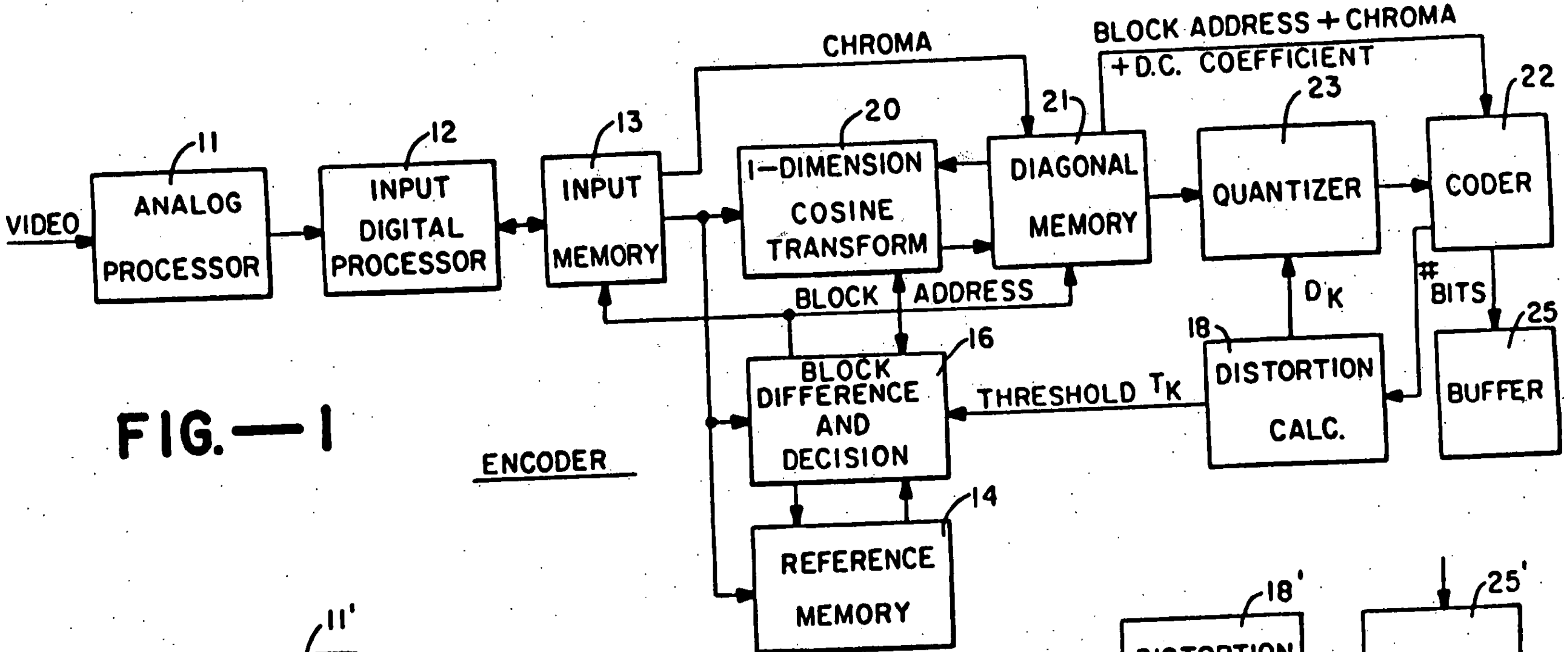
VIDEO
ANALOG PROCESSOR
11
INPUT DIGITAL PROCESSOR
12
INPUT MEMORY
13
CHROMA
1—DIMENSION COSINE TRANSFORM
20
DIAGONAL MEMORY
21
BLOCK ADDRESS + CHROMA
+ D.C. COEFFICIENT
QUANTIZER
23
CODER
22
BITS
BUFFER
25
D_K
DISTORTION CALC.
18
THRESHOLD T_K
BLOCK ADDRESS
BLOCK DIFFERENCE AND DECISION
16
REFERENCE MEMORY
14
ENCODER

FIG.—1

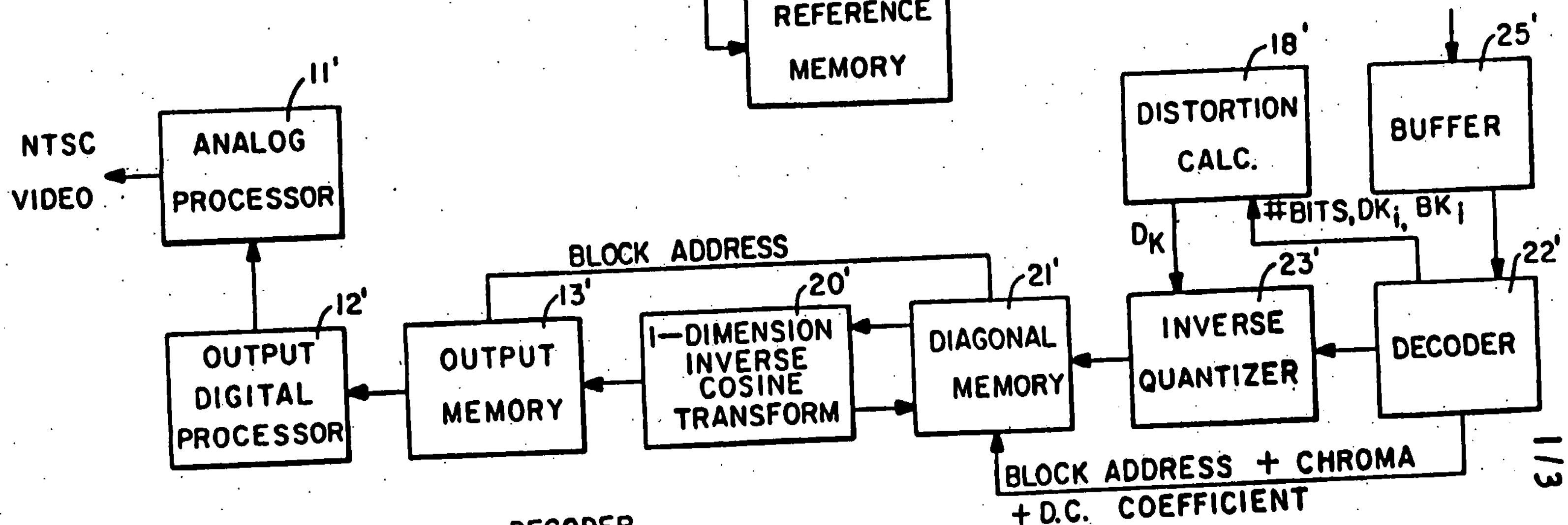
BUFFER
25'
DISTORTION CALC.
18'
#BITS, DK_i, BK_i
D_K
DECODER
22'
INVERSE QUANTIZER
23'
DIAGONAL MEMORY
21'
BLOCK ADDRESS + CHROMA
+ D.C. COEFFICIENT
1—DIMENSION INVERSE COSINE TRANSFORM
20'
BLOCK ADDRESS
OUTPUT MEMORY
13'
OUTPUT DIGITAL PROCESSOR
12'
ANALOG PROCESSOR
11'
NTSC VIDEO
DECODER

FIG.—2

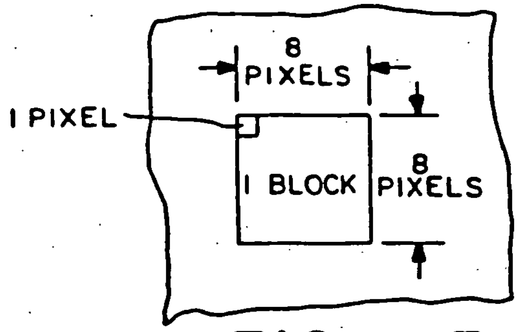

FIG.—3

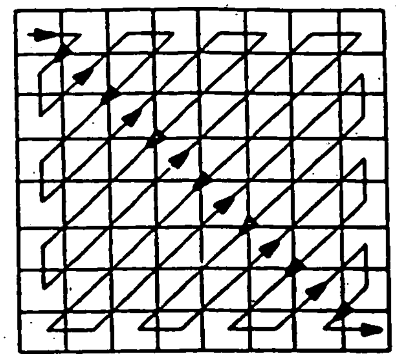

FIG.—5

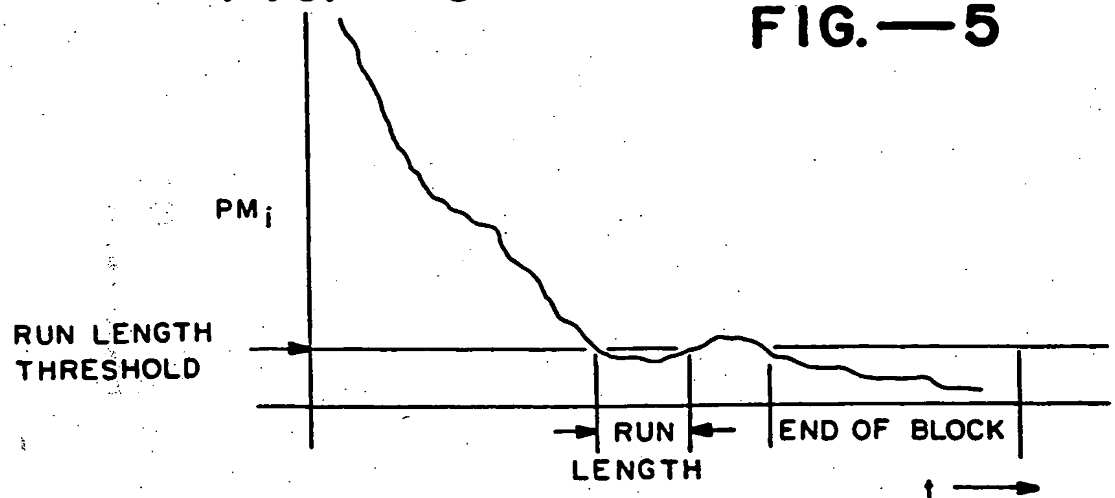

FIG.—7

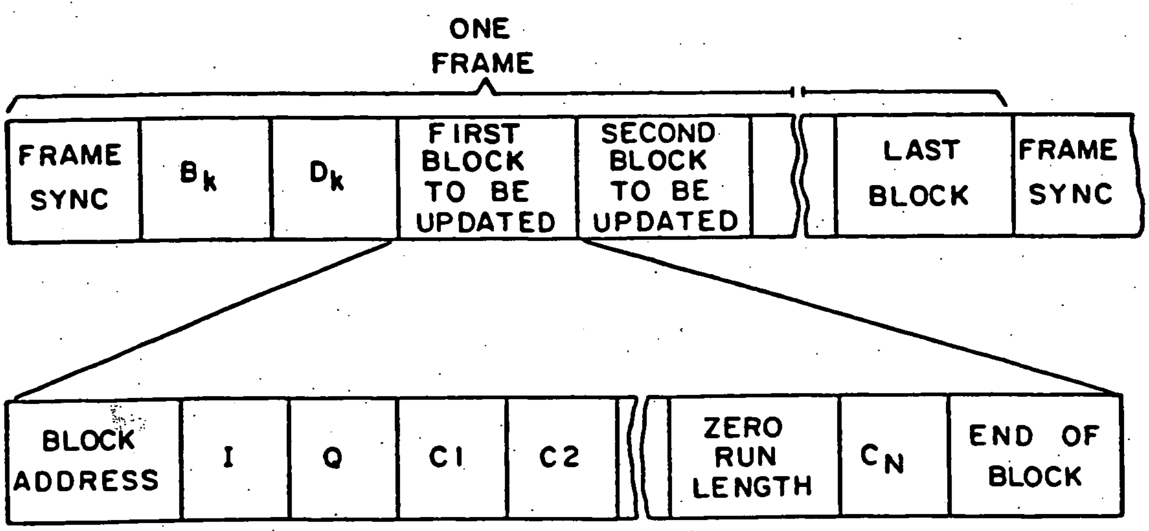

FIG.—8

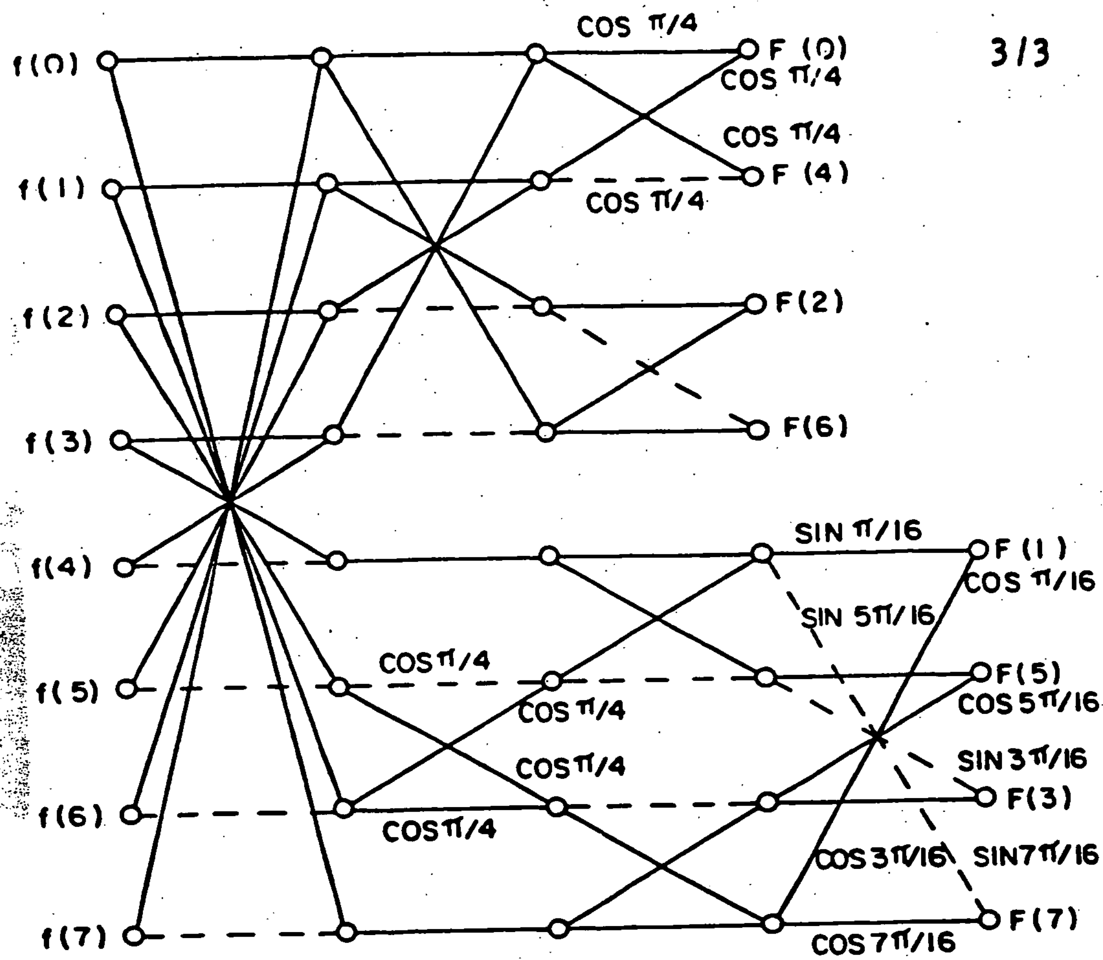
f(0)
f(1)
f(2)
f(3)
f(4)
f(5)
f(6)
f(7)
COS π/4
F (0)
COS π/4
COS π/4
F (4)
COS π/4
F(2)
F(6)
SIN π/16
F(1)
COS π/16
SIN 5π/16
COSπ/4
F(5)
COS 5π/16
COS π/4
COSπ/4
SIN 3π/16
F(3)
COSπ/4
COS 3π/16
SIN7π/16
F(7)
COS 7π/16

FIG.—4

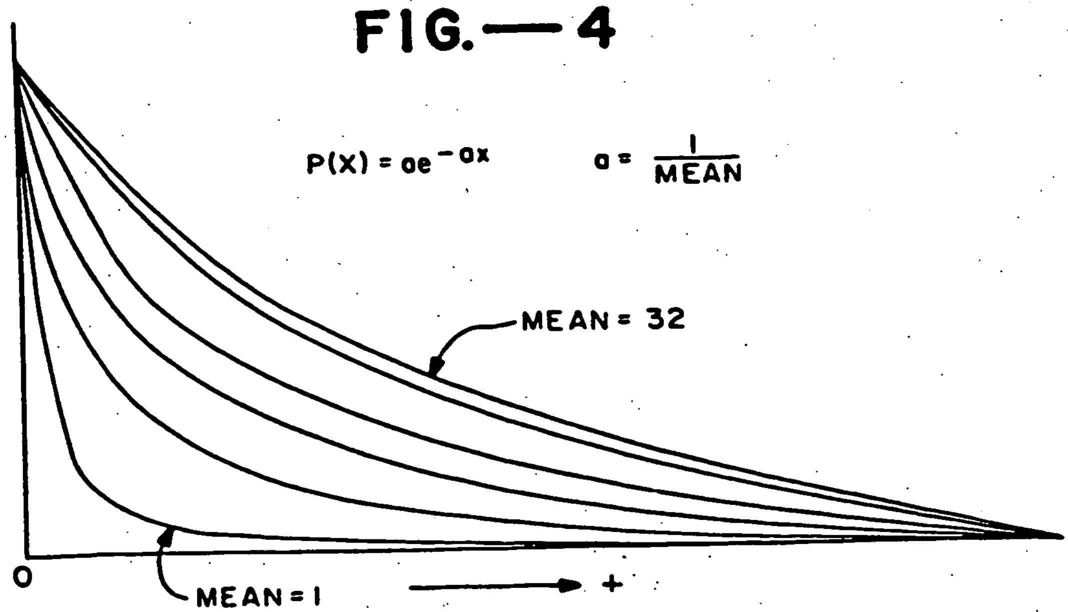
P(X) = ae^{-ax}
a = 1/MEAN
MEAN = 32
0
MEAN = 1
+

FIG.—6

European Patent Office

EUROPEAN SEARCH REPORT

0084270

EP 82307026.3

DOCUMENTS CONSIDERED TO BE RELEVANT			CLASSIFICATION OF THE APPLICATION (Int. Cl. ³)
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	
A	US - A - 4 245 248 (NETRAVALI) * Column 1, lines 14-48 * --		
A	DE - A - 2 336 857 (FUJITSU) * Page 13, claim 1, lines 1-19 * ----		
			TECHNICAL FIELDS SEARCHED (Int. Cl.³)

EPO Form 1503.2 06.78

0084270

European Patent Office

EUROPEAN SEARCH REPORT

EP 82307026.3

DOCUMENTS CONSIDERED TO BE RELEVANT			CLASSIFICATION OF THE APPLICATION (Int. Cl. ³)
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	
A,P	DE - A1 - 3 029 190 (LICENTIA) (18-03-1982) * Fig. 1; page 9, line 4 - page 10, line 12 * --	1,2	H 04 N 7/12
A	IEEE TRANSACTIONS ON COMMUNICATIONS, vol. COM-25, no. 11, November 1977, New York WEN-HSIUNG CHEN et al. "Adaptive Coding of Monochrome and Color Images" pages 1285-1292 * Page 1285, column 1 - column 2, paragraph 2; page 1287, column 1, lines 9-11; page 1288, fig. 6, column 2, lines 1,2; page 1289, column 2, paragraph 1 * --	1,2	TECHNICAL FIELDS SEARCHED (Int. Cl. ³)
A	IEEE TRANSACTIONS ON COMMUNICATIONS, vol. COM-25, no. 11, November 1977, New York ALI HABIBI "Survey of Adaptive Image Coding Techniques" pages 1275-1284 * Page 1278, column 2, paragraph 2, lines 1-8; page 1279, column 1, paragraph 1 * --		H 04 N 1/00 H 04 N 5/00 H 04 N 7/00 H 04 B 1/00 H 04 B 12/00 H 03 K 13/00

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
VIENNA	10-06-1983	BENISCHKA

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document
T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons
& : member of the same patent family, corresponding document

EPO Form 1503 03.82